Claims

What is claimed is:

1.   A system for communicating over a protocol, comprising:

a class factory comprising:
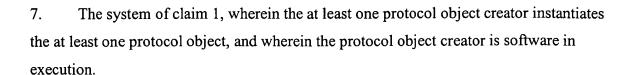
a plurality of identifiers; and

at least one associated registered protocol object creator, the at least one protocol object creator adapted to create at least one protocol object.

2.   The system of claim 1, comprising:

a reading component adapted to read a first data from a resource, the first data having at least one of, a format specific to the protocol and one or more headers and/or footers specific to the protocol when it is read from the resource; and

a writing component adapted to write a second data to the resource, the second data having at least one of, a format specific to the protocol and one or more headers and/or footers specific to the protocol when it is written to the resource.

3.   The system of claim 2, wherein the resource is at least one of, a service, an application and a content source, the resource being accessible over a network.

4.   The system of claim 2, wherein the protocol is at least one of a Hyptertext Transfer Protocol (HTTP), a File Transfer Protocol (FTP) and a Simple Mail Transport Protocol (SMTP).

5.   The system of claim 2, wherein the plurality of identifiers comprise one or more Uniform Resource Identifiers (URIs).

6.   The system of claim 2, wherein the plurality of identifiers comprise one or more prefixes associated with one or more URIs.

7.   The system of claim 1, wherein the at least one protocol object creator instantiates the at least one protocol object, and wherein the protocol object creator is software in execution.

8.   The system of claim 7, wherein the at least one protocol object creator registers one or more implemented creating methods with the class factory, the creating methods being defined in an abstract base class and implemented by the at least one protocol object.

9.   The system of claim 8, wherein the at least one or more protocol object inherits from one or more abstract base classes.

10.   The system of claim 2, wherein the at least one protocol object is adapted to read the first data from the resource.

11.   The system of claim 10, wherein the at least one protocol object makes the data read from the resource available as a byte stream.

12.   The system of claim 11, wherein the at least one protocol object removes at least one of a format specific to the protocol and one or more headers and/or footers specific to the protocol from the first data.

13.   The system of claim 12, wherein the at least one protocol object is adapted to write the second data to the resource.

14.   The system of claim 13, wherein the at least one protocol object accepts a byte stream to write as the second data to the resource.

15.   The system of claim 14, wherein the at least one protocol object adds at least one of a format specific to the protocol and one or more headers and/or footers specific to the protocol to the second data.

16.   The system of claim 1 comprising:

an adding component adapted to add one or more identifiers to a list of registered identifiers and further adapted to add one or more protocol object creating methods to a list of registered protocol object creating methods.

17.   A method for allowing a computer program to communicate over one of a plurality of protocols, comprising:

registering one or more protocol handlers operable to create a protocol object;

receiving a request to communicate;

creating an instance of a protocol object by employing a registered protocol handler;

returning the protocol object to the application; and

using a base class Application Programming Interface (API) to communicate through the protocol object.

18.   The method of claim 17, further comprising:

a computer program generating a request to communicate, wherein the request to communicate identifies a URI with which a communication is desired.

19.   The method of claim 18, wherein creating an instance of a protocol object from a source of registered protocol handlers comprises:

selectively determining one or more protocol object creators operable to create the protocol object based, at least in part, on a portion of the URI, where the protocol object creator implements one or more creator methods defined in an abstract creator base class; and

invoking at least one of the one or more protocol object creators to create the protocol object.

20. The method of claim 19, wherein using the base class API to communicate through the protocol object comprises:

implementing one or more methods defined in the base class API in a derived class; and

employing the one or more implemented methods.

21. The method of claim 20, wherein at least one of the one or more methods can be employed to read a first data from a resource, the first data having at least one of a format specific to the protocol and one or more headers and/or footers specific to the protocol when it is read from the resource, the first data being provided to the computer program as a byte stream.

22. The method of claim 21, wherein at least one of the one or more methods can be employed to write a second data to the resource, the second data having at least one of a format specific to the protocol and one or more headers and/or footers specific to the protocol when it is written to the resource, the second data being provided to the one or more methods as a byte stream.

23. A computer readable medium having computer executable instructions operable to perform the method of claim 17.

24. The computer readable medium of claim 23, further having computer executable instructions operable to perform the method of claim 25.

25. A data packet adapted to be transmitted between two or more computer processes, the data packet comprising:

information operable to facilitate selecting a protocol object creator.

26.   A data packet adapted to be transmitted between two or more computer processes, the data packet comprising:

byte stream data produced by a protocol object, the byte stream data having at least one of a format specific to a protocol and one or more headers and/or footers specific to the protocol removed from a first data read from a resource.

27.   The data packet of claim 26, further comprising:

a second data having at least one of a format specific to the protocol and one or more headers and/or footers specific to the protocol added to a byte stream provided by a computer program.

28.   A system for simplifying application program communication over a protocol, comprising:
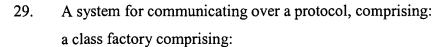
storing means for storing a data related to resolving a Uniform Resource Identifier;

registering means for registering a protocol object creator;

creating means for creating a protocol object;

determining means for selectively determining means for creating a protocol object;

accessing means for accessing a method in the protocol object, which method implements a method defined in a network object base class; and

communicating means for communicating with a resource, wherein the communicating means employ one or more methods in the protocol object.

29.   A system for communicating over a protocol, comprising:

a class factory comprising:

a plurality of identifiers, where the identifiers are URIs; and

at least one registered protocol object creator, the at least one protocol object creator adapted to create at least one protocol object;

at least one protocol object comprising:

a reader adapted to read a first data from a resource, where the first data has formatting specific to the protocol when the first data is read from the resource, where the protocol is at least one of a Hyptertext Transfer Protocol (HTTP), a File Transfer Protocol (FTP) and a Simple Mail Transport Protocol (SMTP), where the resource is at least one of a service, an application and a content source, and where the resource is accessible over a network; and

a writer adapted to write a second data to the resource, where the second data has formatting specific to the protocol when the second data is written to the resource.